Machine Learning Engineer Nanodegree

Capstone Proposal

Samrat Pandiri
March 23rd 2018

TalkingData AdTracking Fraud Detection Challenge

Domain Background

With the latest web technologies there are many new avenues that are opened for the mankind. There are Millions of companies across the Globe that are providing varied services including Domain Registration, Online Gaming, Web Hosting, Cab Booking, Food Ordering, Health Consultation, Online Advertising and lot more. Many of these services makes our life very easy but at the same time we can easily become a victim of a fraud if we are not careful. One of the frauds that is related to Online Advertising is "Click Fraud". Click fraud is a type of fraud that occurs on the Internet in Pay-Per-Click (PPC), Pay-Per-Action (PPA) or Cost-Per-Activity (CPA) online advertising. In this type of advertising, the websites owners that post the ads are paid an amount of money determined by how many visitors to the sites or mobile app click on the ads. Fraud occurs when a person, automated script or computer program imitates a legitimate user clicking on such an ad without having an actual interest in the target of the ad's link.

Problem Statement

One of the major areas of "Click Fraud" is in the area of mobile ad channels where automated scripts may click mobile ads or download a mobile app without a real reason. The problem that we will be solving here is to predict if a user click is genuine or fraudulent. With over 1 billion smart mobile devices in active use every month, China is the largest mobile market in the world and therefore suffers from huge volumes of fraudulent traffic.

TalkingData, China's largest independent big data service platform, covers over 70% of active mobile devices nationwide. They handle 3 billion clicks per day, of which 90% are potentially fraudulent. Their current approach to prevent click fraud for app developers is to measure the journey of a user's click across their portfolio, and flag IP addresses who produce lots of clicks, but never end up installing apps.

So, here we have to build an algorithm that predicts whether a user will download an app after clicking a mobile app Ad.

Datasets and Inputs

To build an algorithm that predicts whether a user will download an app after clicking a mobile app ad TalkingData has provided us a generous dataset covering approximately 200 million clicks over 4 days.

File descriptions

- train.csv - The training set consisting of approximately 200 Million rows
- train_sample.csv - 100,000 randomly-selected rows of training data
- test.csv - the testing set consisting of approximately 20 Million rows

Data fields

Each row of the training data contains a click record, with the following features.

- ip: IP Address of click.
- app: App id for marketing.
- device: Device type id of user mobile phone (e.g., iPhone 6 Plus, iPhone 7, Huawei mate 7, etc.)
- os: OS version id of user mobile phone
- channel: Channel id of mobile ad publisher
- click_time: Timestamp of click (UTC)
- attributed_time: If user download the app for after clicking an ad, this is the time of the app download
- is_attributed: The target that is to be predicted, indicating the app was downloaded

Solution Statement

We will build a solution for the above stated problem and try to identify click frauds using Machine Learning. As we have an imbalanced data set, we will try to build multiple models and then do a Stacking Classifier or a Voting Classifier. For building the basic models, we will use algorithms like LightGBM or XGBoost. Then a custom built Stacking Classifier or a pre-defined Voting Classifier will be used to get better predictions.

Benchmark Model

For the benchmark model we will use a random forest algorithm on a sample dataset. As the dataset is very huge we will try to build out benchmark model using 5% to 10% of the training data instead of the whole 200 Million data points. Based on the benchmark score we will build a model that does better than the benchmark model by a reasonable margin.

We will also use the " random_forest_benchmark" that is already present on the Kaggle Leaderboard with a score of 0.9117.

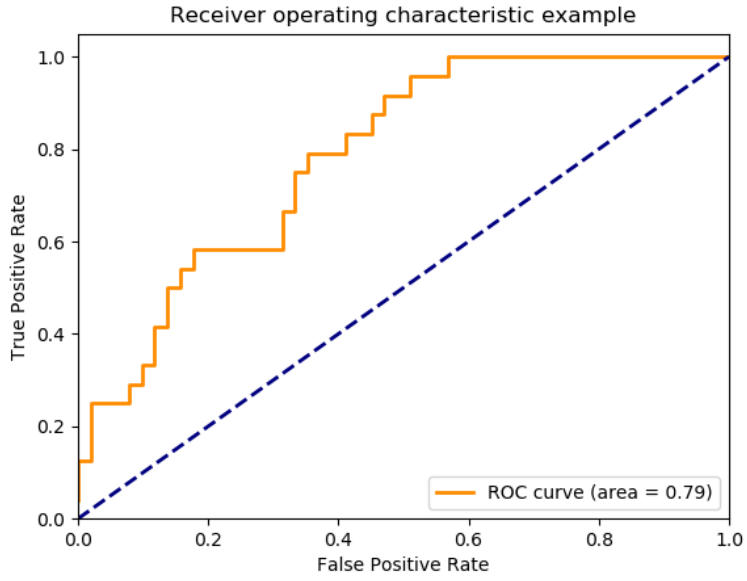
Evaluation Metrics

As the submissions are evaluated on area under the ROC curve between the predicted probability and the observed target we will also use the same metric.

A receiver operating characteristic (ROC), or simply ROC curve, is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied. It is created by plotting the fraction of true positives out of the positives (TPR = true positive rate) vs. the fraction of false positives out of the negatives (FPR = false positive rate), at various threshold settings. TPR is also known as sensitivity, and FPR is one minus the specificity or true negative rate.

We will use the ***roc_auc_score*** in sklearn that computes the area under the receiver operating characteristic (ROC) curve, which is also denoted by AUC or AUROC.

By computing the area under the roc curve, the curve information is summarized in one number.



Project Design

We will use the below design flow to solve the above mentioned problem.

- As we have a huge data set with around 200 Million rows, care should be taken to load the data in parts so that no memory errors are seen.
- Basic data cleaning will be done to remove any unnecessary columns and fill NaN with meaningful data.
- EDA and Data Visualization will be done to get a better understanding of the data.
- Preprocessing will be done on the data as per the requirements of the machine learning algorithms that will be chosen.
- A benchmark model will be built on the sample data.
- Couple of model based on different algorithms like LightGBM or XGBoost will be built.
- A Voting Classifier or Stacking Classifier will be used on the base models to get better predictions.
- Hyper-Parameter tuning will be done to get better predictions.

References:

- https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection
- https://en.wikipedia.org/wiki/Click_fraud
- https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection/data
- https://github.com/Microsoft/LightGBM
- http://xgboost.readthedocs.io/en/latest/
- https://mlwave.com/kaggle-ensembling-guide/
- http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html
- https://en.wikipedia.org/wiki/Receiver_operating_characteristic
- https://www.youtube.com/watch?v=OAl6eAyP-yo
- https://www.coursera.org/learn/machine-learning-projects